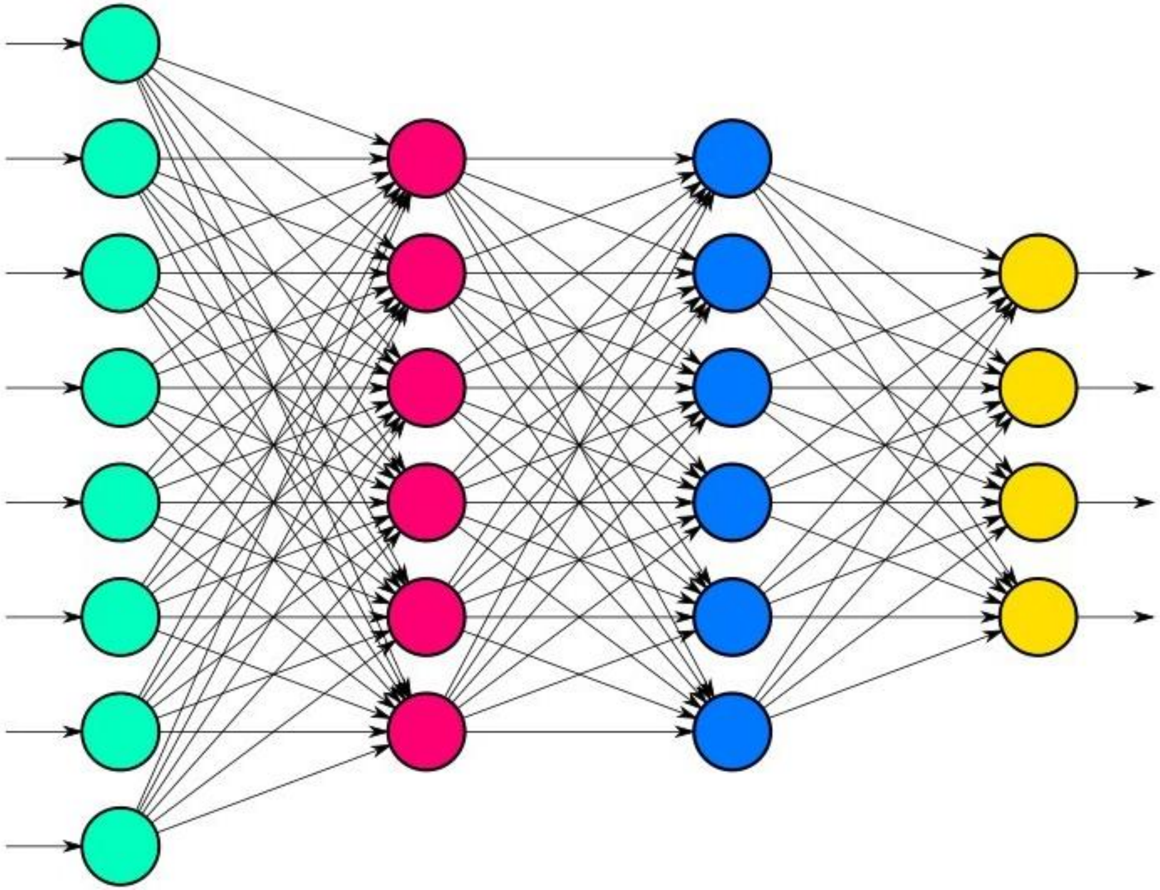
Redes Neuronales Recurrentes

Julio Waissman Vilanova

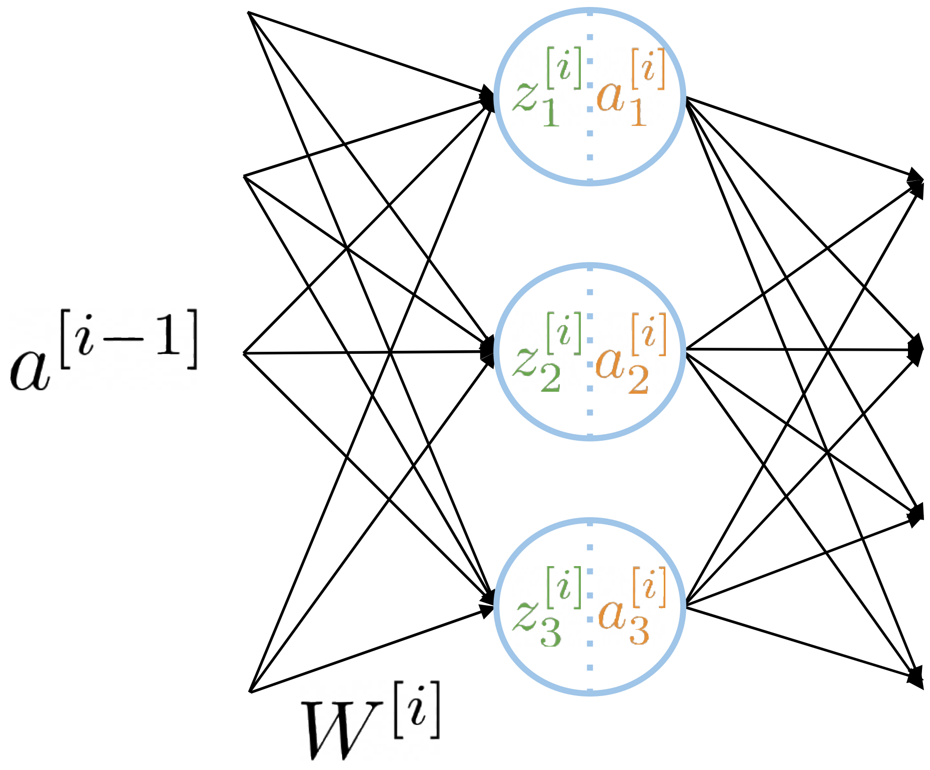
Mayo, 2024

MCD

Recordando las redes neuronales



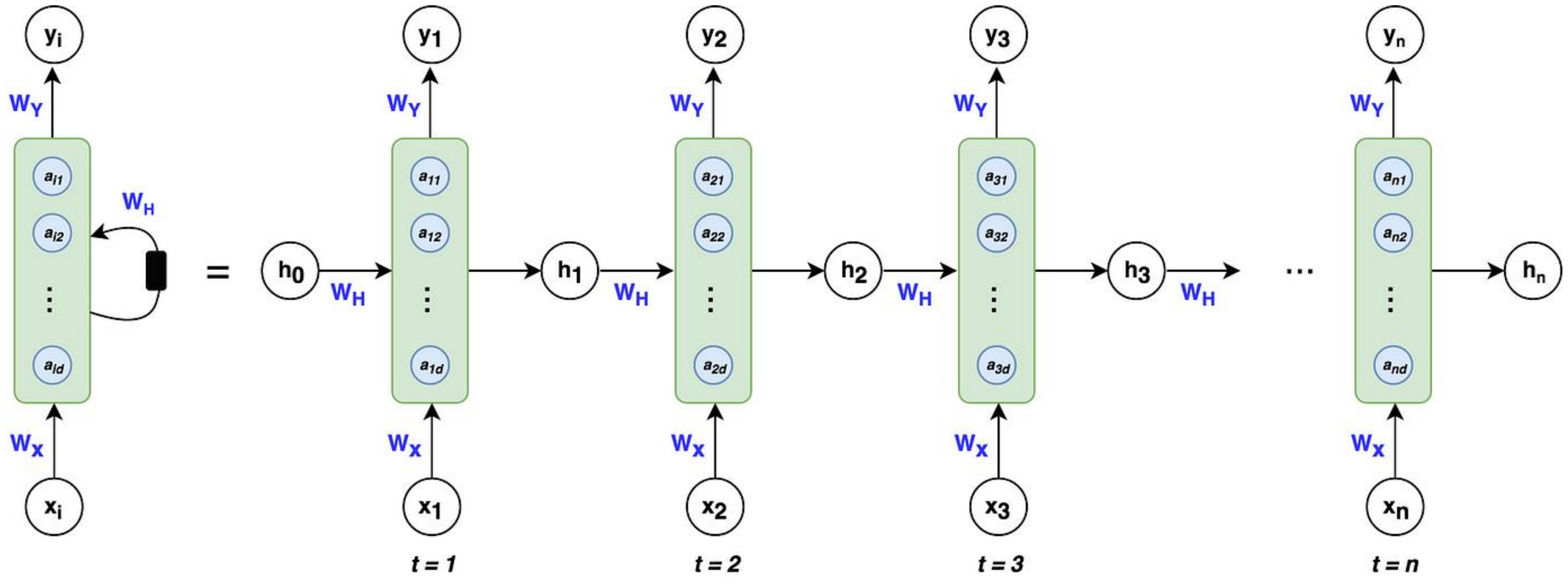
Capas densas



Dense Layer $\longrightarrow$ $z^{[i]} = W^{[i]} a^{[i-1]}$

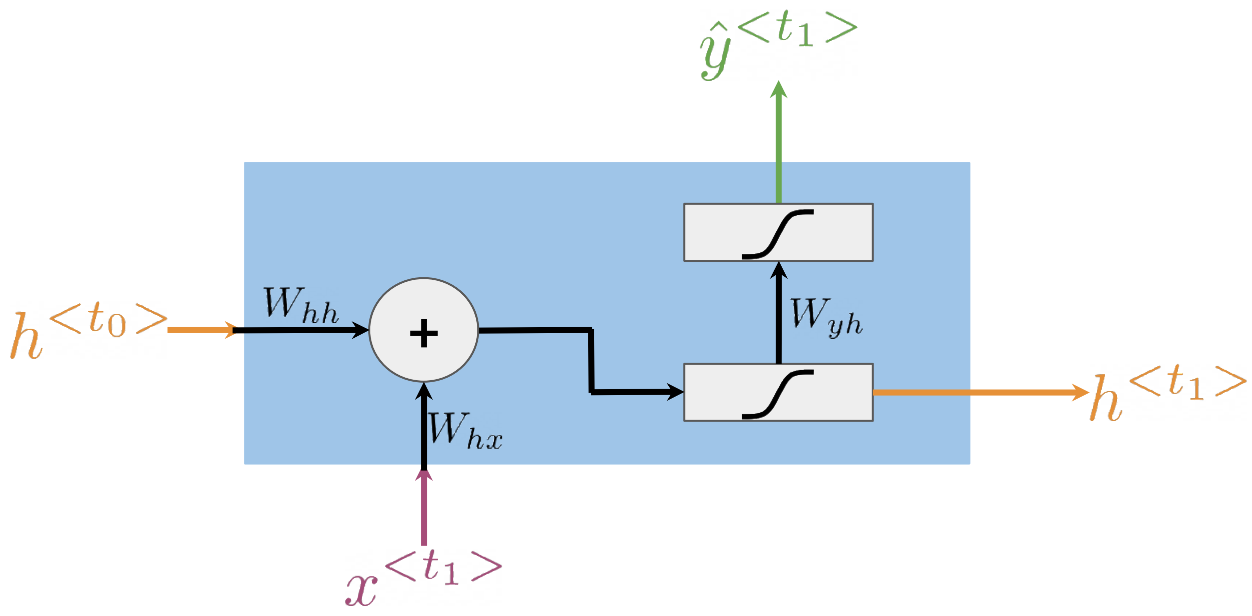
ReLU Layer $\longrightarrow$ $g(z^{[i]}) = \max(0, z^{[i]})$

Redes recurrentes sencillas

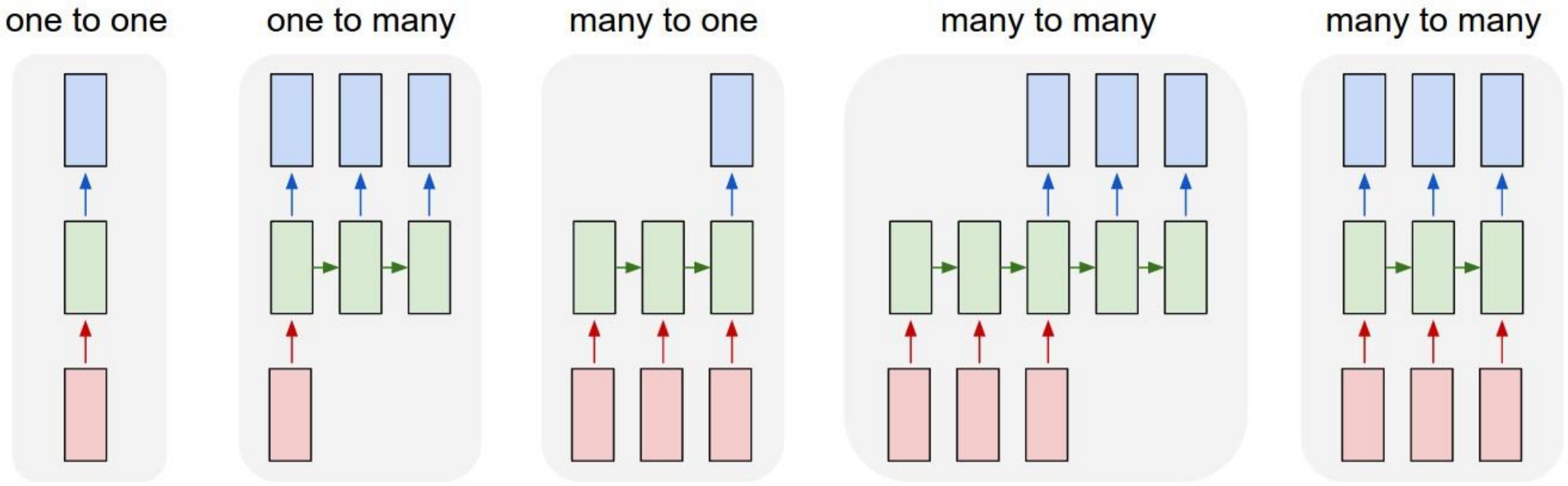


$$h_t = f(W_x x_t + W_h h_{t-1} + b_h)$$
$$y_t = g\left(W_y h_t + b_y\right)$$

Arquitectura de una res recurrente sencilla

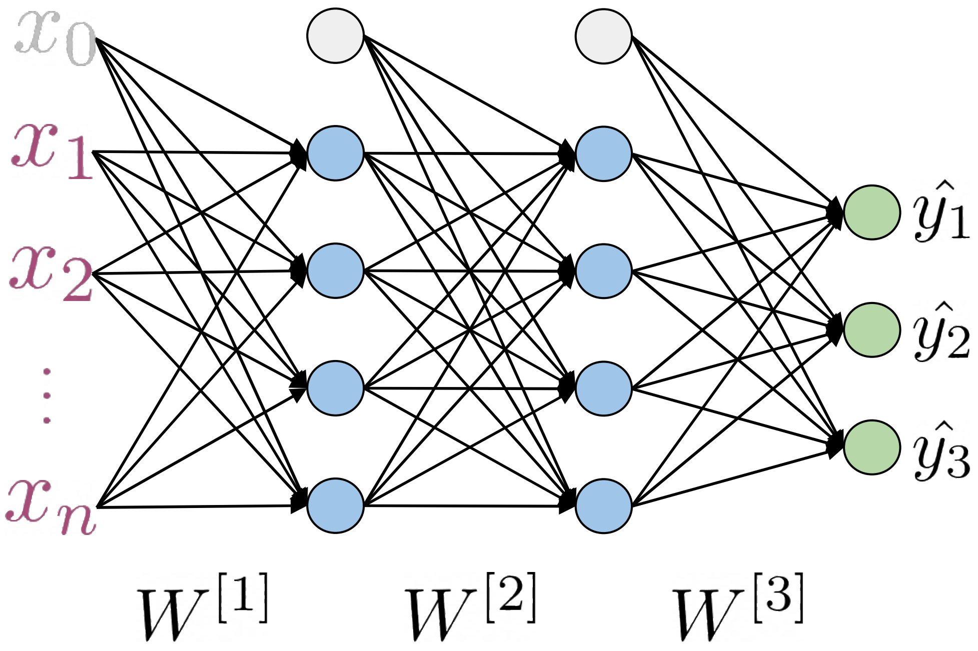


Tipos de problemas a resolver con RNN



Aprendizaje en redes neuronales

Cross Entropy Loss



K - classes or possibilities

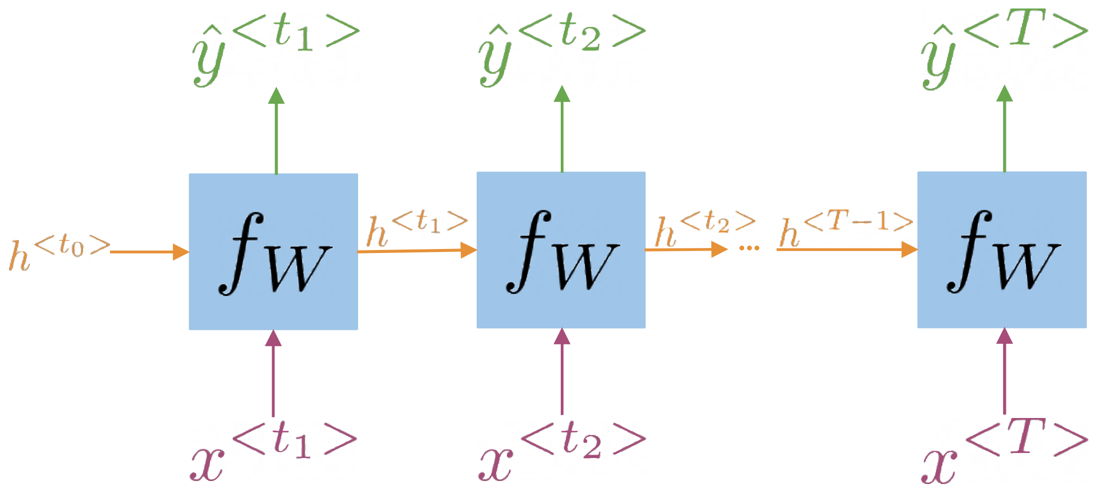
Either 0 or 1

$$J = -\sum_{j=1}^{\boxed{K}} \boxed{y_j} \log \hat{y}_j$$

Looking at a single example (x, y)

Generalización a una RNN

Cross Entropy Loss



$$h^{<t>} = g(W_h[h^{<t-1>}, x^{<t>}] + b_h)$$

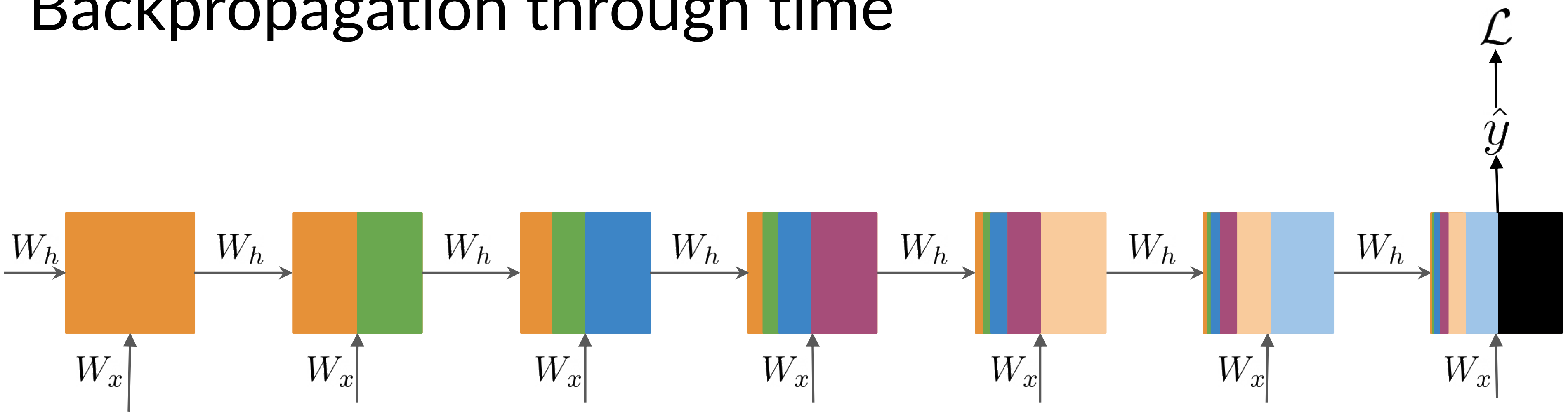
$$\hat{y}^{<t>} = g(W_{yh} h^{<t>} + b_y)$$

$$J = -\frac{1}{T} \sum_{t=1}^{T} \sum_{j=1}^{K} y_j^{<t>} \log \hat{y}_j^{<t>}$$

Average with respect to time

Aprendizaje en una RNN: BPTT

Backpropagation through time



W_x
W_h
Same at every step

$$\frac{\partial L}{\partial W_h} \propto \sum_{1 \leq k \leq t} \left(\prod_{t \geq i > k} \frac{\partial h_i}{\partial h_{i-1}} \right) \frac{\partial h_k}{\partial W_h}$$

Gradient is proportional to a sum of partial derivative products

Aprendizaje en una RNN: BPTT

Backpropagation through time

$$\frac{\partial L}{\partial W_h} \propto \sum_{1 \leq k \leq t} \boxed{\left(\boxed{\prod_{t \geq i > k} \frac{\partial h_i}{\partial h_{i-1}}} \right) \frac{\partial h_k}{\partial W_h}}$$

Contribution of hidden state ***k***

Length of the product proportional to how far ***k*** is from ***t***

$$\boxed{\frac{\partial h_t}{\partial h_{t-1}} \frac{\partial h_{t-1}}{\partial h_{t-2}} \frac{\partial h_{t-2}}{\partial h_{t-3}} \frac{\partial h_{t-3}}{\partial h_{t-4}} \frac{\partial h_{t-4}}{\partial h_{t-5}} \frac{\partial h_{t-5}}{\partial h_{t-6}} \frac{\partial h_{t-6}}{\partial h_{t-7}} \frac{\partial h_{t-7}}{\partial h_{t-8}} \frac{\partial h_{t-8}}{\partial h_{t-9}} \frac{\partial h_{t-9}}{\partial h_{t-10}} \frac{\partial h_{t-10}}{\partial W_h}}$$

Contribution of hidden state ***t-10***

Aprendizaje en una RNN: BPTT

Backpropagation through time

$$\frac{\partial L}{\partial W_h} \propto \sum_{1 \le k \le t} \boxed{\left(\boxed{\prod_{t \ge i > k} \frac{\partial h_i}{\partial h_{i-1}}} \right) \frac{\partial h_k}{\partial W_h}}$$

Contribution of hidden state ***k***

Length of the product proportional to
how far ***k*** is from ***t***

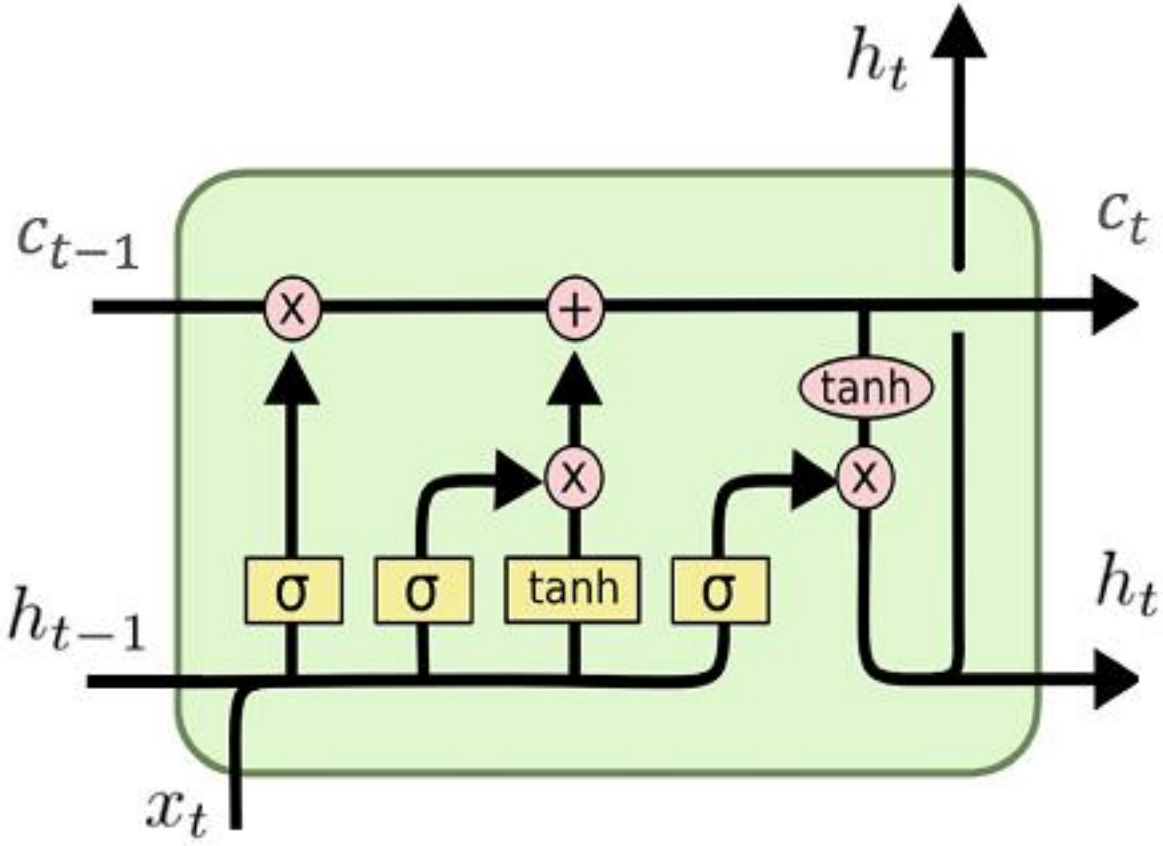
Partial derivatives <1	Contribution goes to 0	Vanishing Gradient
Partial derivatives >1	Contribution goes to infinity	**Exploding** Gradient

LSTMs: Una solución memorable

- Aprende cuando recordar u cuando olvidar
- Se compone de:
  - Un estado de celda (*cell state*)
  - Un estado oculto (*hidden state*)
  - Multiples compuertas

Las compuertas evitan que explote o desvanezca el gradiente en BPTT

Arquitectura de una celda LSTM



$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$$

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$$

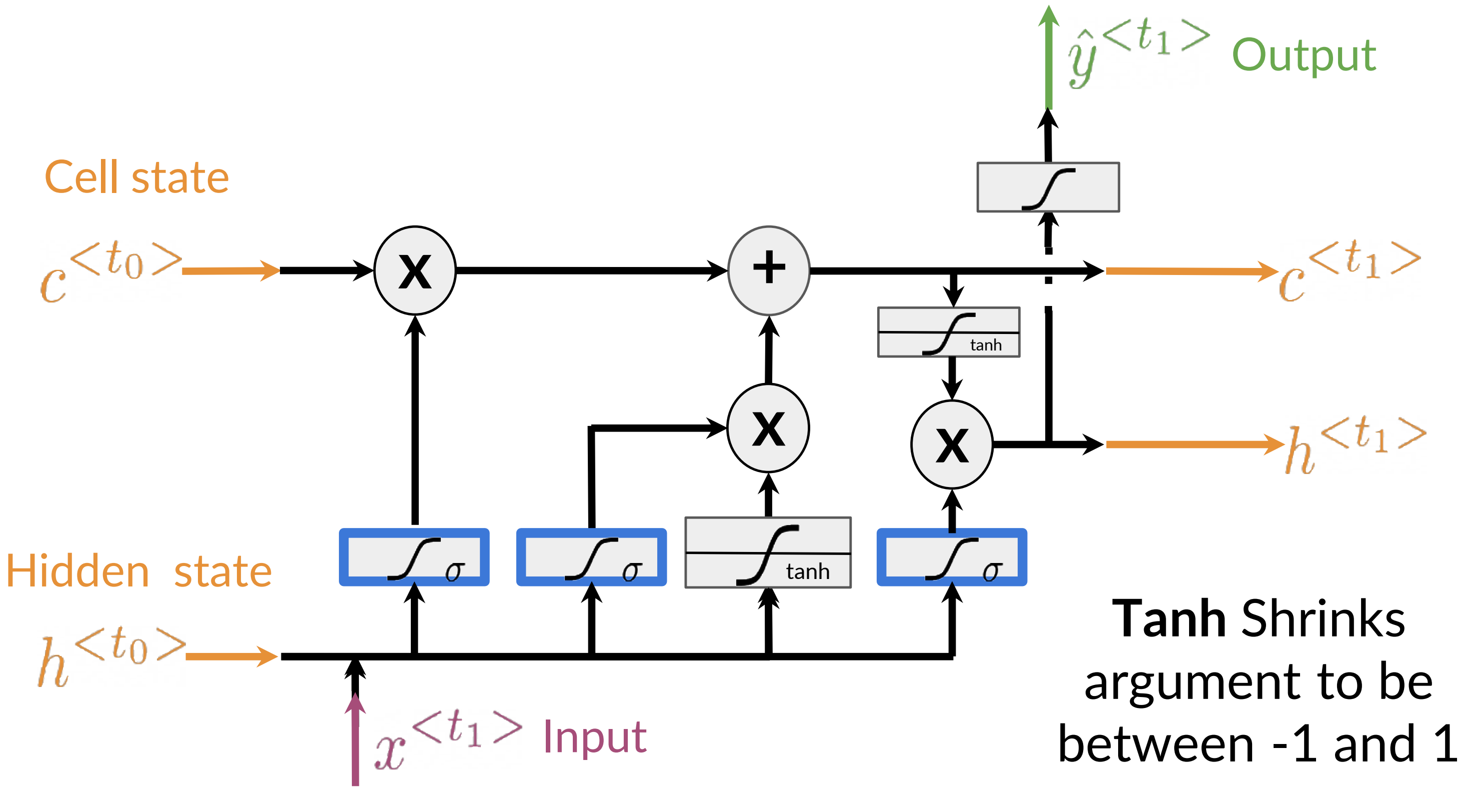
$$\tilde{c}_t = tanh(w_c[h_{t-1}, x_t] + b_c)$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$$

$$h_t = o_t * tanh(c^t)$$

Arquitectura de una celda LSTM

Candidate Cell State



Candidate cell state

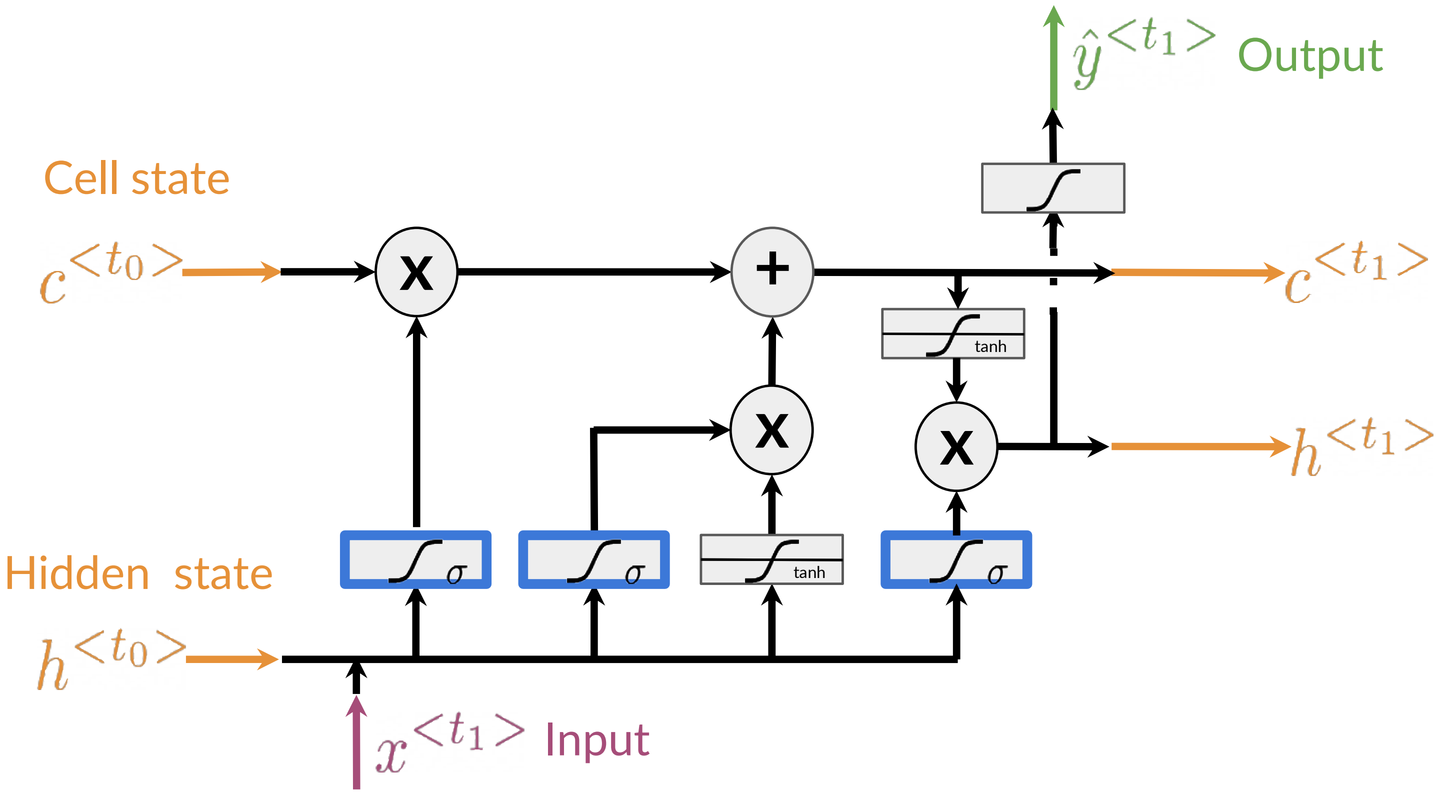
Information from the previous **hidden state** and current **input**

Tanh Shrinks argument to be between -1 and 1

Other activations could be used

Arquitectura de una celda LSTM

New Cell State

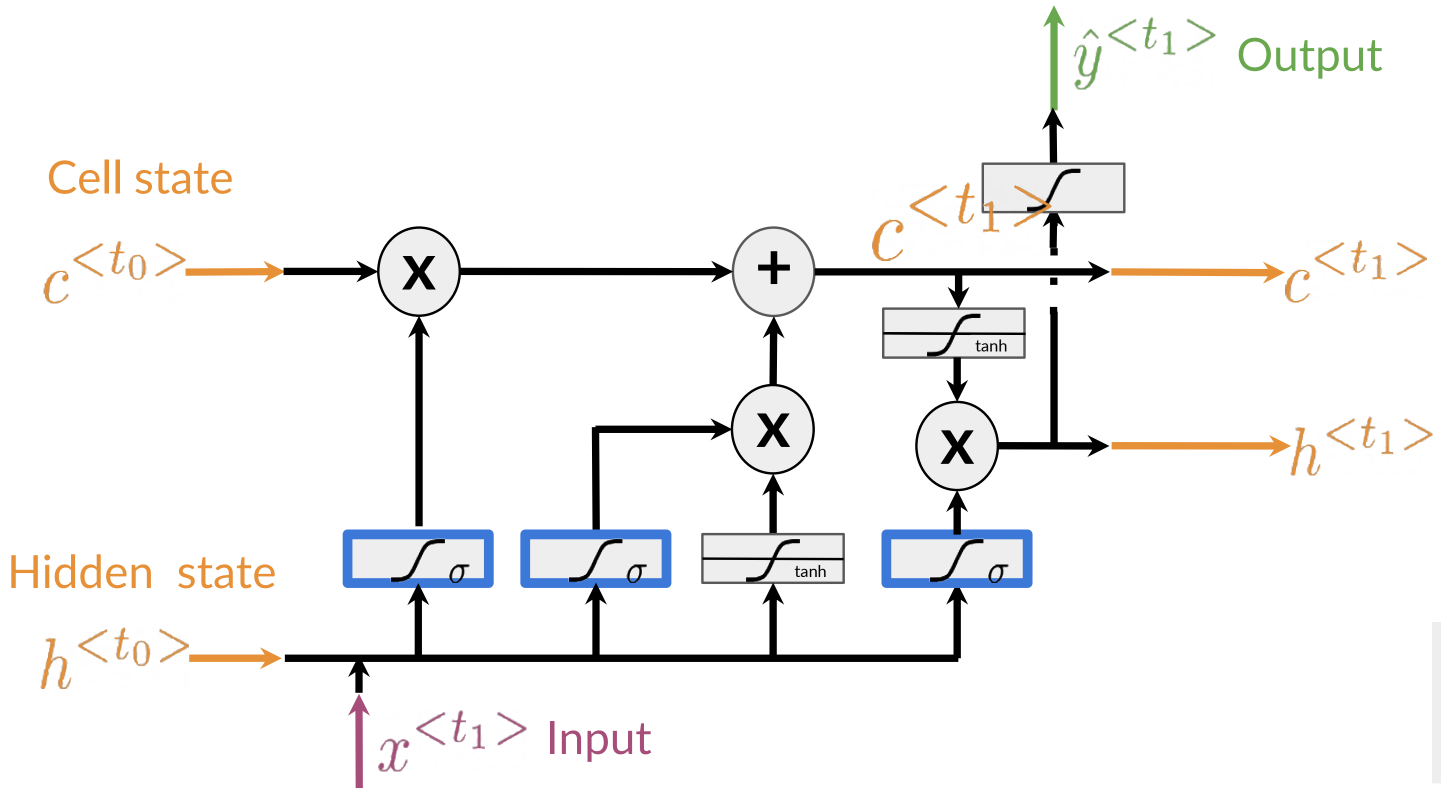


New Cell state

Add information from the **candidate cell state** using the **forget** and **input gates**

Arquitectura de una celda LSTM

New Hidden State

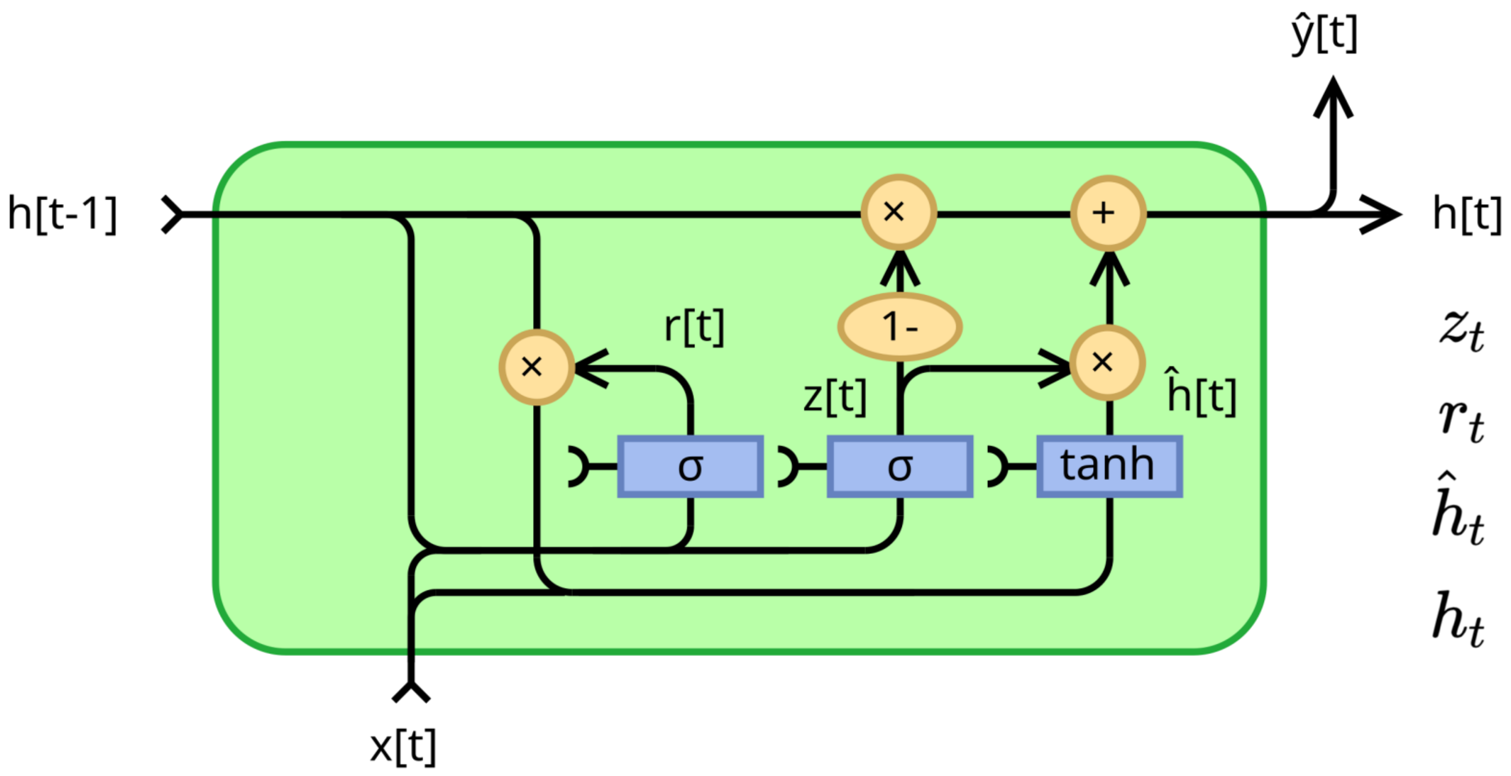


New Hidden State

Select information from the **new cell state** using the **output gate**

The **Tanh** activation could be omitted

Arquitectura GRU



$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$
$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$
$$\hat{h}_t = \phi(W_h x_t + U_h (r_t \odot h_{t-1}) + b_h)$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t$$

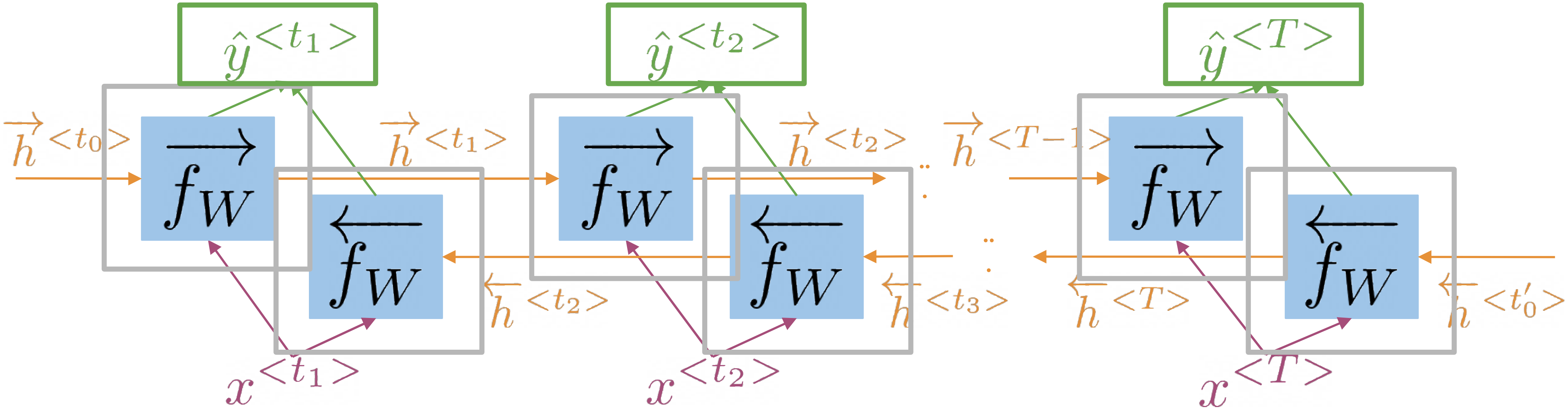
Una celda LSTM simplificada

Redes recurrentes bidireccionales: Motivación

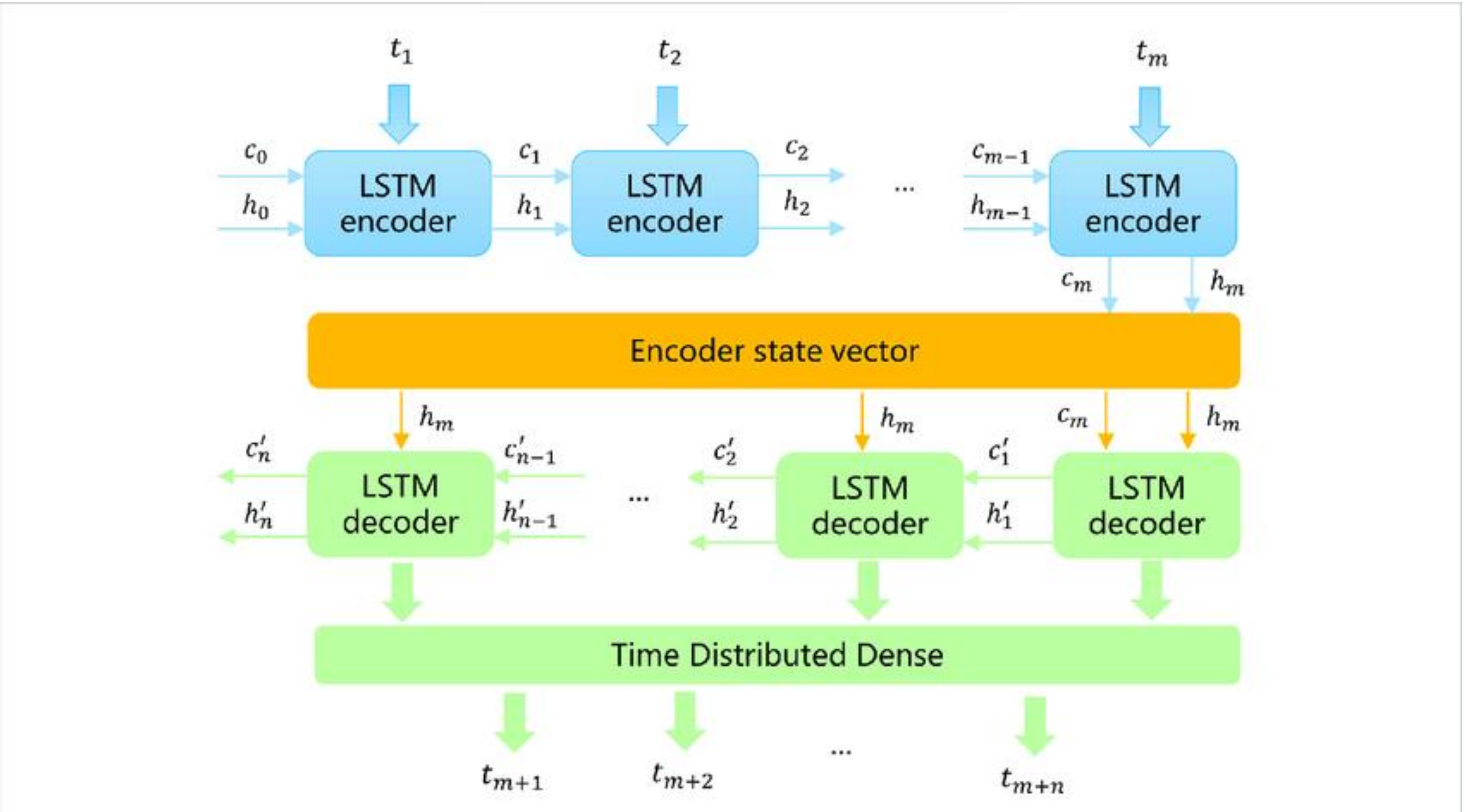
Le marqué, pero ___ no contesta el teléfono. Yo creo que a Elaine no le gusta que le hablen”

- Necesidad de conocer el texto completo para resolver el problema.
- El problema es secuencial, pero se puede asumir un conocimiento de la secuencia completa de entrada.

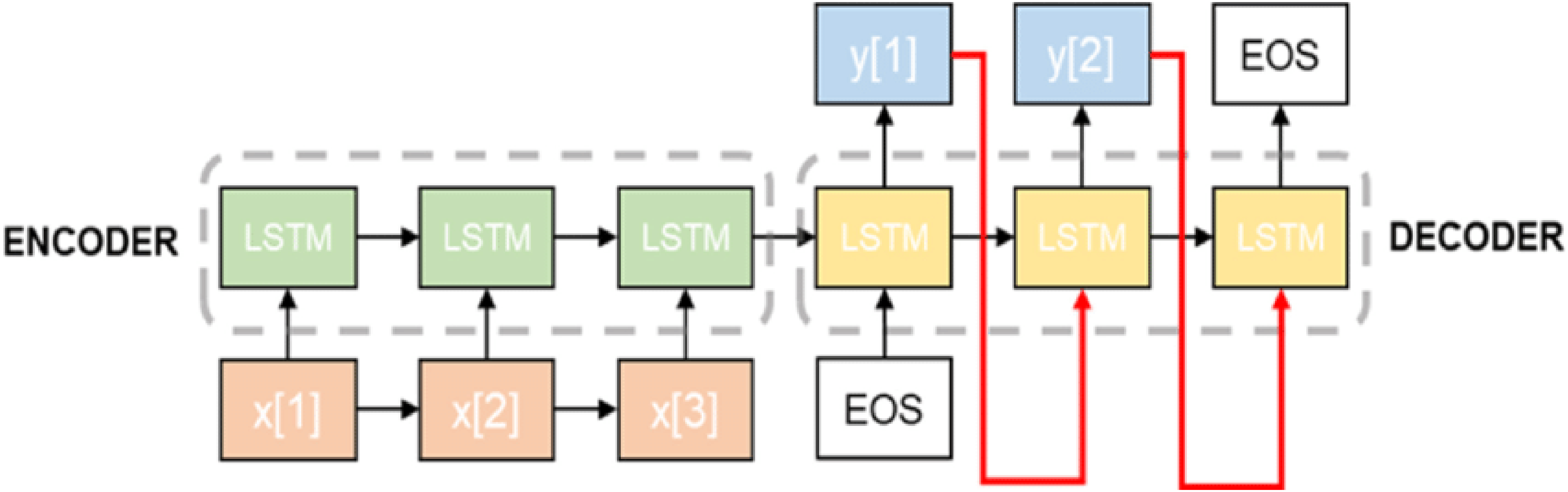
Redes recurrentes bidireccionales



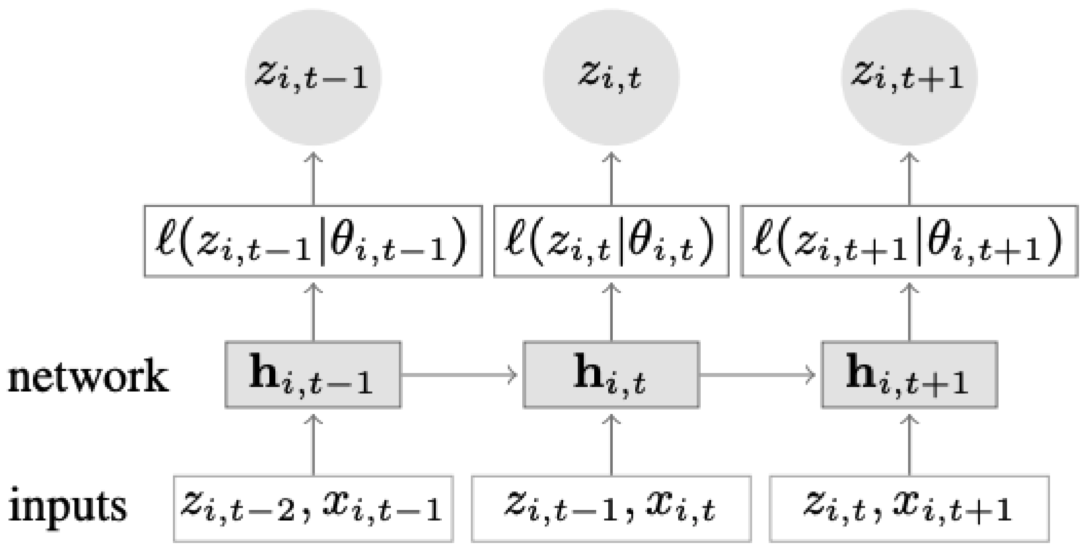
Modelos Seq2seq



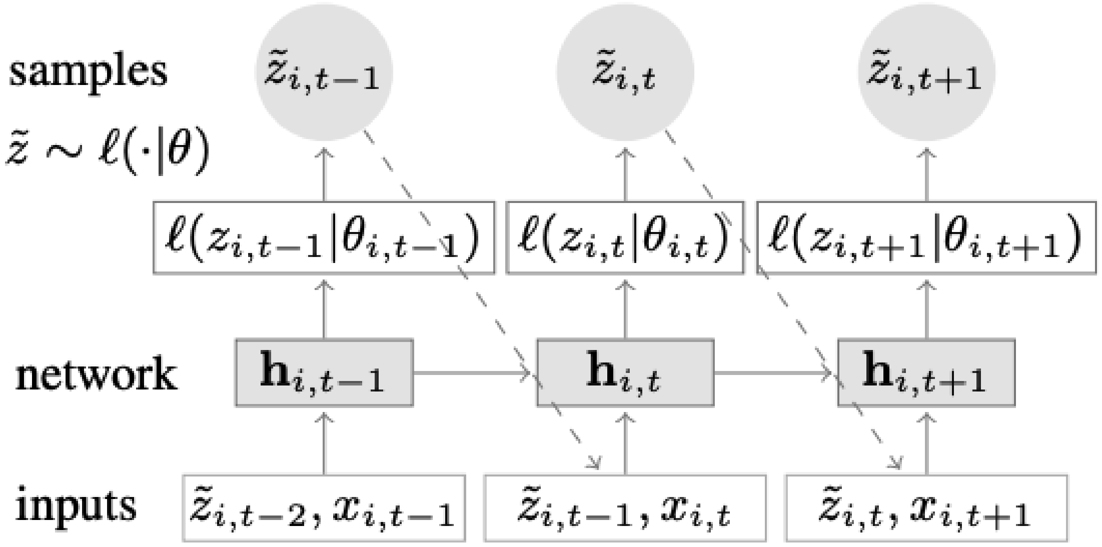
Modelos Seq2seq



DeepAR

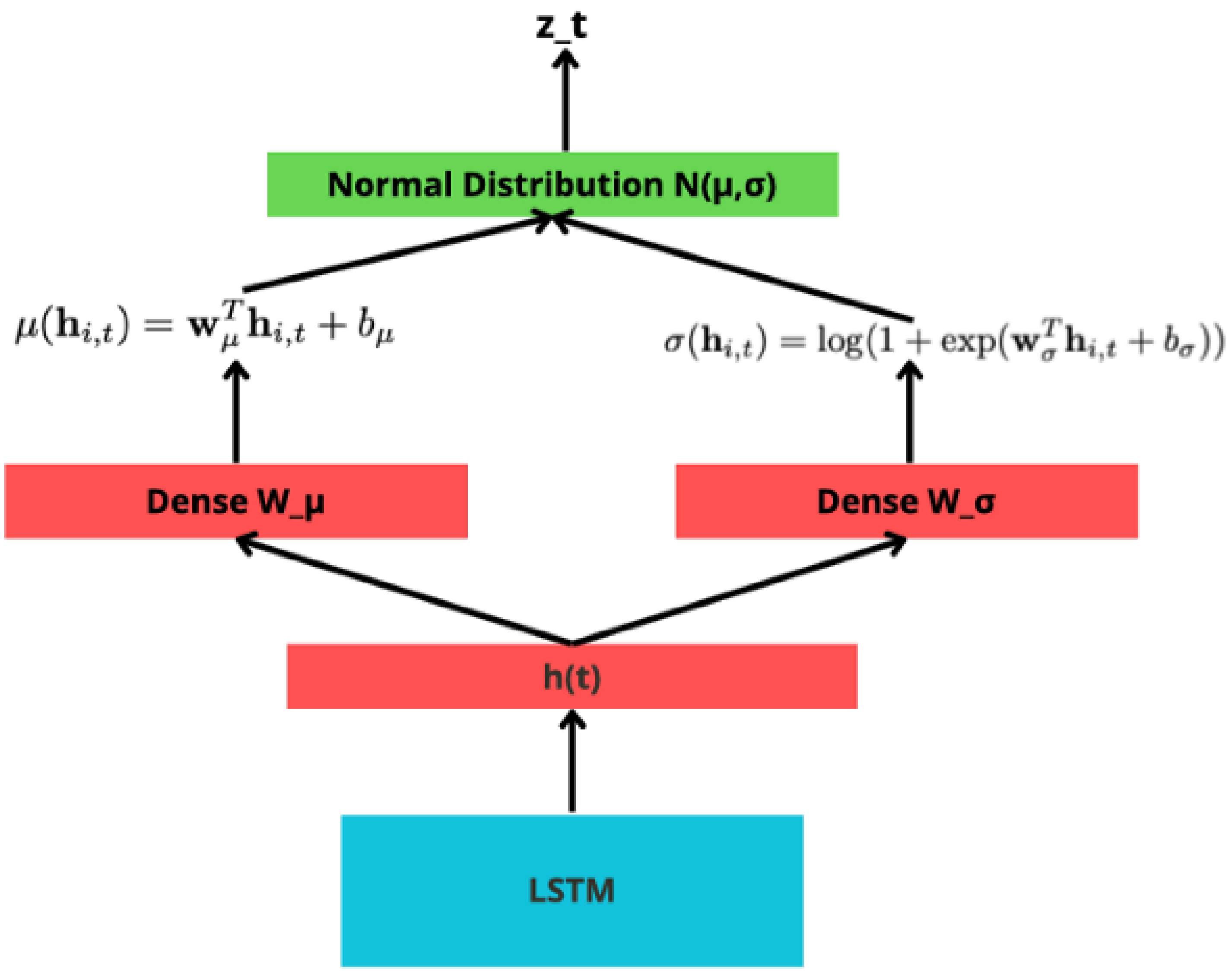


Entrenamiento



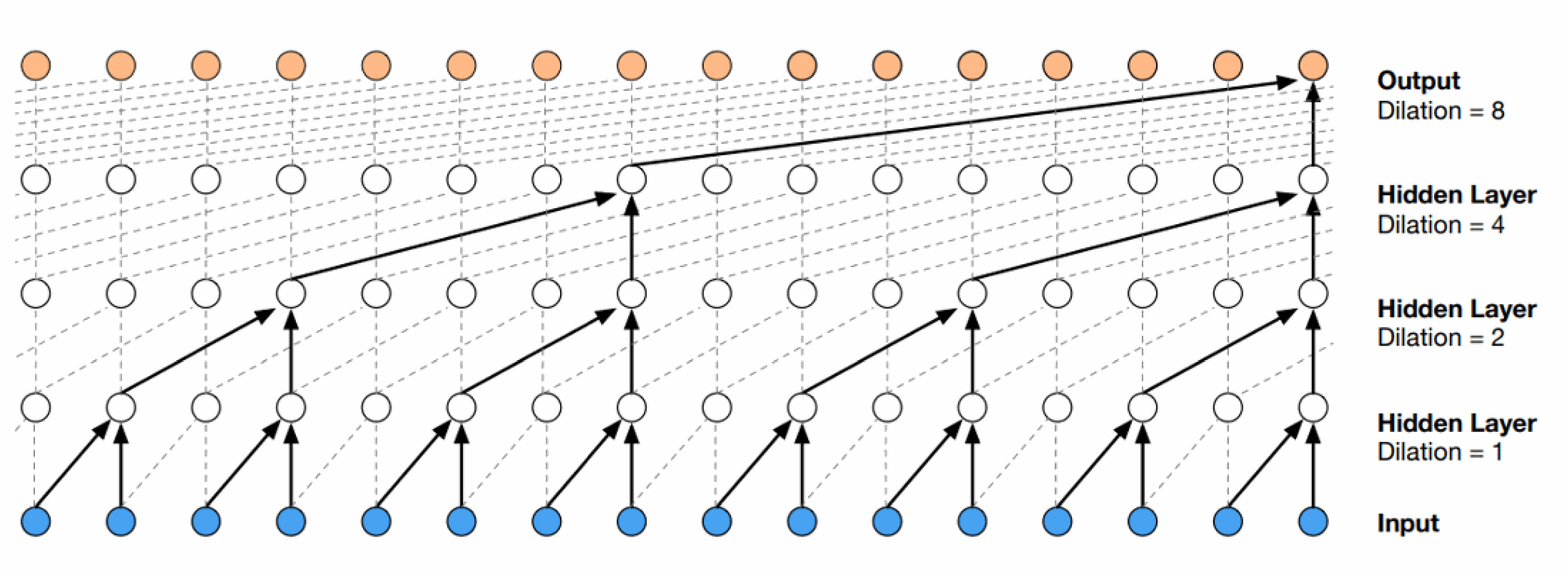
Inferencia

DeepAR



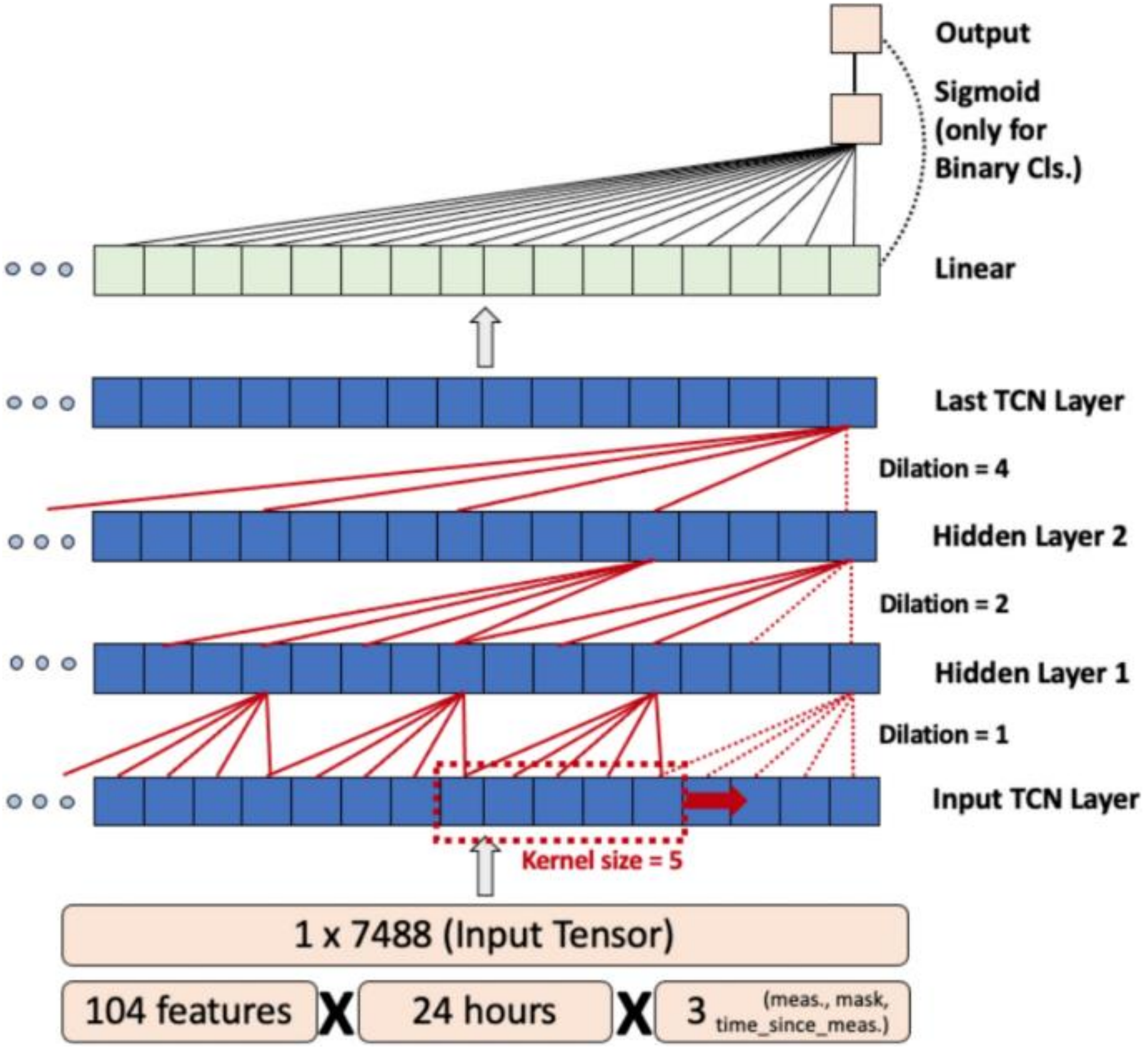
z_t
Normal Distribution N(μ,σ)
$\mu(\mathbf{h}_{i,t}) = \mathbf{w}_\mu^T \mathbf{h}_{i,t} + b_\mu$
$\sigma(\mathbf{h}_{i,t}) = \log(1 + \exp(\mathbf{w}_\sigma^T \mathbf{h}_{i,t} + b_\sigma))$
Dense W_μ
Dense W_σ
h(t)
LSTM

TCN



Output
Dilation = 8
Hidden Layer
Dilation = 4
Hidden Layer
Dilation = 2
Hidden Layer
Dilation = 1
Input

TCN



Temporal convolutional networks and data rebalancing for clinical length of stay and mortality prediction